Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-13 (2018) 443–449

Predefined-time convergence in task-based inverse dynamics using time base generators

Gustavo Arechavaleta *, Jonathan Obregón *,
Héctor M. Becerra **, América Morales-Díaz *

* *Robótica y Manufactura Avanzada, CINVESTAV, Unidad Saltillo,
Av. Industria Metalúrgica 1062, Ramos Arizpe, Coah., C.P. 25900,
(e-mail: {garechav,jonathan.obregon,america.morales}@cinvestav.mx.)*
** *Centro de Investigación en Matemáticas, Guanajuato 36023, México
(e-mail: hector.becerra@cimat.mx)*

Abstract: In this paper, we propose a state-feedback controller that is capable to achieve predefined-time convergence for a class of nonlinear second order systems that can be linearized to a perturbed double integrator. The controller relies on the use of time base generators. Also, we study the link between the proposed predefined-time controller and the widely applied task-based inverse dynamics framework to control kinematically redundant robots. We show that positioning tasks assigned to the robot subject to joint-limits constraints are successfully completed in a user defined time regardless of the robot's initial state. Moreover, the proposed task-based controller is robust to parametric uncertainties and matched disturbances since it includes a term given by a super-twisting control. This is verified through simulations with a kinematically redundant mobile-manipulator KUKA youBot with eight degrees of freedom.

Keywords: predefined-time convergence, robot control in task space, inverse dynamics.

1. INTRODUCTION

Many applications in robotics require generating and controlling dynamically feasible motion tasks subject to a strict time schedule. This drives the need for control design techniques capable of ensuring a desired convergence time. In this sense, many finite-time and fixed-time control schemes have been proposed in the last years (see for example Levant (2013) and Polyakov et al. (2015)), however neither of both approaches guarantee convergence in a specific (predefined) time. Although fixed-time is harder than finite-time convergence, the former only guarantees the existence of a bound for the convergence time that is independent of the initial conditions (Bhat and Bernstein, 1998; Polyakov, 2012).

Recent work has demonstrated that, for nonlinear systems that can be linearized to n-order integrator systems, methods based on time base generators can play an important role in controlling robots (Becerra et al., 2017). A major advantage of this control framework over finite-time and fixed-time controllers is that the accomplishment of the control objectives is ensured in a desired time defined by the user regardless of the initial conditions, which is called *predefined-time convergence* (Torres et al., 2015; Jimenez-Rodriguez et al., 2017). The scheme introduced in Becerra et al. (2017) takes advantage of parametric time signals that converge to zero in a specific time, i.e. time base generators (TBG), and they can be tracked using feedback controllers.

Our main observation is that the predefined-time convergence property of control systems is compatible with recent advances in task-based robot control, in particular hierarchical inverse dynamics (Saab et al., 2013). This approach, which more explicitly controls each task coordinate expressed in terms of the robot's state, allows the execution of multiple tasks by exploiting the kinematic redundancy of the robot (Samson et al., 1991; Khatib, 1987). The possible conflicts among tasks are handled by means of a hierarchical structure (Siciliano and Slotine, 1991). However, the evaluation of tasks within the control-loop heavily relies on the computation of the robot's equations of motion (Nakanishi et al., 2008). Thus, the resulting control input usually suffers from the well-known problem of model uncertainties. Some strategies have been suggested to cope with this problem such as the scheme in Chang and Jeong (2012).

In this work, we construct a control law in the task space that is robust to bounded disturbances while the predefined-time convergence of each task is guaranteed independently of the robot's initial state. Disturbances usually come from uncertainties in masses and inertias associated to the robot model. Thus, the predefined-time convergence is addressed as the stabilization of the error between the task variables and TBG signals, robustified with a super-twisting control (STC) term. To apply the STC to second order systems, we define a sliding surface such that it has relative degree one.

The rest of the paper is organized as follows: Section 2 presents the case of predefined-time convergence of second order systems. Section 3 introduces the task-based inverse dynamics. Section 4 presents the integration of the predefined-time controller in the hierarchical inverse dynamics scheme. Section 5 details

2405-8963 © 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.07.318

simulation results for a mobile manipulator KUKA youBot. Section 6 concludes with some remarks.

2. PREDEFINED-TIME CONVERGENCE OF SECOND ORDER SYSTEMS

The aim of this section is to design a time-varying state-feedback control law that achieves predefined-time convergence for nonlinear second order systems that can be linearized to double integrator systems of the form

$$\begin{aligned}\dot{e}_1 &= e_2\\ \dot{e}_2 &= u(t) + \rho(t)\\ y &= a_1 e_1 + a_2 e_2\end{aligned} \tag{1}$$

where $\rho(t)$ is a bounded matched disturbance, $u(t)$ and y are the system input and output, respectively. The initial conditions are $e_1(t_i) = e_{1i}$ and $e_2(t_i) = e_{2i}$ where t_i is the initial time.

Let us define the desired final time as t_f. Then, the predefined-time convergence property of a control law implies that the settling time, $t_s = t_f - t_i$, is specified while guaranteeing *constant convergence time* independently of initial conditions. For that, we define the following time-dependent polynomial functions known as *time base generators* (TBG):

$$h_j(t) = \begin{cases} g(t)\cdot c_j & \text{if } t \in [t_i, t_f] \\ 0 & \text{otherwise.}\end{cases} \tag{2}$$

where $j \in \{1,2\}$, $g(t) = [t^r \; t^{r-1} \; \ldots \; t \; 1]$ is the time basis vector with $r \geq 5$ and c_k is a vector of coefficients. The functions $h_j(t)$ are continuous and smooth such that $\dot{h}_j(t)$ and $\ddot{h}_j(t)$ exist. In addition, the conditions over (2) with respect to t_i and t_f are:

$$\begin{aligned} h_1(t_i) = \dot{h}_2(t_i) &= 1\\ h_2(t_i) = \dot{h}_1(t_i) &= 0\\ h_j(t \geq t_f) = \dot{h}_j(t \geq t_f) &= 0\end{aligned} \tag{3}$$

2.1 Computation of TBG functions

A straightforward method to compute $h_j(t)$ and its first and second time derivatives consists of solving the following linear system to find the coefficients c_j for $r = 5$:

$$[c_1 \;\, c_2] = \begin{bmatrix} G(t_i)\\ G(t_f)\end{bmatrix}^{-1} \begin{bmatrix} I \\ 0\end{bmatrix} \tag{4}$$

where $c_j \in \mathbb{R}^6$ is the vector of coefficients,

$$G(t) = \begin{bmatrix} g(t)\\ \dot{g}(t)\\ \ddot{g}(t)\end{bmatrix}$$

is the time basis matrix, $I \in \mathbb{R}^{3\times 2}$ is an identity matrix and $0 \in \mathbb{R}^{3\times 2}$ a matrix of zeros. It is, however, possible to allow more flexibility during the computation of coefficients by setting $r > 5$ as it has been suggested in Becerra et al. (2017). In that case, a simple convex quadratic program solves the problem.

2.2 A robust predefined-time control law

The objective now is to use the TBG as reference profiles in a tracking control scheme. In particular, we take advantage of both the TBG together with a *super-twisting controller* (STC), which is capable to cope with matched uncertainties/disturbances $\rho(t)$ according to Moreno and Osorio (2012). Let us define the tracking error as

$$\begin{aligned}\varepsilon_1(t) &= e_1(t) - h_1(t)e_{1i} - h_2(t)e_{2i}\\ \varepsilon_2(t) &= e_2(t) - \dot{h}_1(t)e_{1i} - \dot{h}_2(t)e_{2i}\end{aligned} \tag{5}$$

Theorem 1. Consider the functions (2) together with their first time-derivative subject to (3) and the tracking errors (5). There exist gains $\{k_1, k_2, k_{fr}\} \in \mathbb{R}$ such that the continuous time-variant STC law

$$\begin{aligned} u &= \ddot{h}_1(t)e_{1i} + \ddot{h}_2(t)e_{2i} - k_{fr}\left(e_2(t) - e_2^d(t)\right)\\ &\quad -k_1|s|^{1/2}\text{sign}(s) + v,\\ \dot{v} &= -k_2\text{sign}(s),\end{aligned} \tag{6}$$

with $s = k_{fr}\left(e_1(t) - e_1^d(t)\right) + e_2(t) - e_2^d(t)$, achieves predefined-time convergence for the disturbed second order system (1) where the desired references are

$$\begin{aligned} e_1^d(t) &= h_1(t)e_{1i} + h_2(t)e_{2i}\\ e_2^d(t) &= \dot{h}_1(t)e_{1i} + \dot{h}_2(t)e_{2i}\end{aligned}$$

Moreover, global asymptotic stability of (5) is attained.

Proof. The stability proof of the robust predefined-time controller (6) for the perturbed double integrator (1) is a particular instance of a more general proof given in Becerra et al. (2017) since it has been proved for n-order integrator systems. The main idea is to use the behavior of the error dynamics,

$$\begin{aligned}\dot{\varepsilon}_1 &= \varepsilon_2\\ \dot{\varepsilon}_2 &= u + \rho - \ddot{h}_1(t)e_{1i} - \ddot{h}_2(t)e_{2i},\end{aligned} \tag{7}$$

together with the time-derivative of the sliding surface

$$\dot{s} = k_{fr}\left(e_2(t) - e_2^d(t)\right) + u + \rho - \ddot{h}_1(t)e_{1i} - \ddot{h}_2(t)e_{2i}, \tag{8}$$

and the control law (6) to get the closed-loop dynamics

$$\begin{aligned}\dot{s} &= -k_1|s|^{1/2}\text{sign}(s) + \sigma\\ \dot{\sigma} &= -k_2\text{sign}(s) + \dot{\rho}\end{aligned} \tag{9}$$

where $\sigma = v + \rho$. The remaining of the proof relies on the fact that for an appropriate choice of control gains k_1 and k_2, the global convergence in finite time of (9) to the origin, i.e. $s = \sigma = 0$, is ensured in the presence of continuously and smooth bounded disturbance $|\rho| < L$ and $|\dot{\rho}| < M$ for some constants $L > 0$ and $M > 0$ as it has been proved in Moreno and Osorio (2012). Knowing that (7) is constrained to $s = \dot{s} = 0$, from the definition of the sliding surface we have that $\varepsilon_2 = \dot{\varepsilon}_1 = -k_{fr}\varepsilon_1$. It implies the global asimptotic stability of the tracking errors (5). Consequently, the state of the system (1) tracks the TBG references and it is driven to the origin in the predefined time window t_s.

3. TASK-BASED INVERSE DYNAMICS

The robot inverse dynamics problem involves the computation of the robot's equations of motion. For this purpose, we use the d'Alembert-Lagrange formulation:

$$A(q)\ddot{q} + b(q,\dot{q}) = \tau \tag{10}$$

where $\{q, \dot{q}, \ddot{q}\} \in \mathbb{R}^n$ are the joint position, velocity and acceleration, respectively. $A(q) \in \mathbb{R}^{n\times n}$ denotes the symmetric positive definite inertial matrix, the vector of nonlinear terms is:

$$b(q,\dot{q}) \triangleq C(q,\dot{q})\dot{q} + g(q) \tag{11}$$

where $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ contains the Coriolis and centrifugal terms, $g(q) \in \mathbb{R}^n$ represents the gravity and $\tau \in \mathbb{R}^n$ is the generalized input torques.

A task is simply an error function in terms of the robot's configuration together with the corresponding differential mapping between the task and configuration coordinates associated to the robotic system:

$$e = x(q) - x_d \in \mathbb{R}^m \tag{12}$$

where $x(q)$ is obtained by means of forward kinematics, x_d is the desired value of the task. We assume that (12) is twice differentiable with respect to time:

$$\ddot{e} = J\ddot{q} + \dot{J}\dot{q} \tag{13}$$

where $J = \dfrac{\partial e}{\partial q} \in \mathbb{R}^{m\times n}$ is the task Jacobian. By solving for $\ddot{q}$ in (10) and plugging it in (13) yields:

$$\ddot{e} = JA^{-1}\tau + \mu \tag{14}$$

where $\mu = -JA^{-1}b + \dot{J}\dot{q}$ is the drift of the task. The task-based inverse dynamics is obtained by solving for τ in (14) as follows:

$$\tau = \left(JA^{-1}\right)^{\#_A}(\ddot{e} - \mu) \tag{15}$$

where $\left(JA^{-1}\right)^{\#_A} = J^T\left(JA^{-1}J^T\right)^{-1}$ is the weighted generalized inversion of JA^{-1}.

The hierarchical task-based inverse dynamics is a powerful control framework to exploit the kinematic redundancy of robotic systems. The main idea is to define a set of tasks to be executed simultaneously. This is possible because the dimension of the task space is less than that of the configuration space, i.e. $m < n$. To overcome possible conflicts among tasks, we impose a hierarchy between them such that (15) becomes:

$$\begin{aligned} \tau &= \sum_{i=1}^{p} \tau_i, \qquad\qquad (16)\\ \tau_i &= \left(J_iA^{-1}N_{i-1}\right)_i^{\#_A}\left(\ddot{e}_i - \mu_i - J_iA^{-1}\tau_{i-1}\right),\\ N_i &= N_{i-1} - \left(J_iA^{-1}N_{i-1}\right)_i^{\#_A} J_iA^{-1}N_{i-1} \end{aligned}$$

where $\tau_0 = 0$ and $N_0 = I_n$. Notice that N_i is the recursive null-space projector of $J_iA^{-1}N_{i-1}$.

4. THE PROPOSED CONTROL SCHEME

First, consider a single task such that its dynamics is

$$\ddot{e} = JA^{-1}u_q + \mu \tag{17}$$

with the control law:

$$u_q = \left(\widetilde{JA^{-1}}\right)^{\#_A}(u - \widetilde{\mu}) \tag{18}$$

where $u_q \in \mathbb{R}^n$ is the joint space control input. The modeling uncertainties of the robotic system are contained in $\widetilde{JA^{-1}} \in \mathbb{R}^{m\times n}$ and $\widetilde{\mu} \in \mathbb{R}^m$. The predefined-time controller in task space is $u \in \mathbb{R}^m$, which can be decoupled in m second order systems. Each u_k for $k = 1,\dots,m$, is constructed as (6) to track the individual error dynamics of (17) while fulfilling the predefined-time convergence property. Thus, the closed-loop system has the form:

$$\ddot{e} = JA^{-1}\left(\widetilde{JA^{-1}}\right)^{\#_A} u + \rho \tag{19}$$

where

$$\rho = -JA^{-1}\left(\widetilde{JA^{-1}}\right)^{\#_A}\widetilde{\mu} + \mu \tag{20}$$

is the bounded perturbation. If we consider ideal dynamics, $\rho = JA^{-1}\left(JA^{-1}\right)^{\#_A}\mu - \mu = I_m\mu - \mu = 0$, then (19) becomes $\ddot{e} = u$ and Theorem 1 can be used to stabilize the error.

Suppose now the robot is asked to perform simultaneously p hierarchical tasks. The control law becomes:

$$u_q = \sum_{i=1}^{p} u_{q_i} \tag{21}$$

with

$$u_{q_i} = \left(\widetilde{J}_i\widetilde{A^{-1}}\widetilde{N_{i-1}}\right)^{\#_A}\left(u_i - \widetilde{\mu}_i - \widetilde{J}_i\widetilde{A^{-1}}u_{q_{i-1}}\right) \tag{22}$$

and $u_{q_0} = 0$. The task space control inputs are the vectors $u_i \in \mathbb{R}^{m_i}$ where each $u_{i,k}$ for $k = 1,\dots,m_i$ has the form of (6). Note that the index i enumerates p hierarchical tasks while k refers to an individual control signal for tracking one task coordinate. Therefore, there are $\sum m_i \leq n$ decoupled second order systems, and the structure of the closed-loop system is:

$$\ddot{e}_i = J_iA^{-1}\left(\widetilde{J}_i\widetilde{A^{-1}}\widetilde{N_{i-1}}\right)^{\#_A} u_i + \rho_i \quad \text{for} \quad i = 1,\dots,p \tag{23}$$

with

$$\rho_i = -J_iA^{-1}\left(\widetilde{J}_i\widetilde{A^{-1}}\widetilde{N_{i-1}}\right)^{\#_A}\widetilde{\mu}_i + \mu_i - J_iA^{-1}u_{q_{[\setminus i]}} \tag{24}$$

where the first and second terms on the right-hand side of (24) encode the parametric uncertainties with respect to the drift of the i-th task. The third term on the right-hand side of (24) can be seen as matched disturbances given by the contribution of the other hierarchical tasks to perform the i-th task. The control input $u_{q_{[\setminus i]}}$ denotes (21), but it excludes the task i. Thus, ρ_i plays the role of the bounded perturbation in the system (1). If two or more tasks are considered, (23) introduces a multiplicative term to the control input that might degenerate the performance of the TBG tracking by applying directly the robust control law of Theorem 1. We have assumed that such factor tends to be a diagonal matrix, and its effect can be compensated by the control gains. However, further investigation is needed to verify that assumption. In this sense, promising results have been obtained in simulations as reported in Section 5. The hierarchical structure handles conflicts between tasks of the form (12). Therefore, the convergence to zero of lower hierarchical tasks is subject to the appropriate convergence of tasks with higher hierarchy in the predefined time window t_s.

5. SIMULATION RESULTS

We implemented the torque-mode predefined-time controller (21) on a mobile manipulator KUKA youBot in MATLAB R2016b. The robot is composed of eight degrees of freedom:

$$q = \begin{bmatrix} q_b \\ q_m \end{bmatrix} \in SE(2) \times \mathbb{T}^5 \tag{25}$$

where $q_b \in SE(2)$ corresponds to the position and orientation coordinates of the vehicle, and $q_m \in \mathbb{T}^5$ is the manipulator's joint coordinates. The generalized torques are obtained from wheel's and manipulator torques $\tau_b \in \mathbb{R}^4$ and $\tau_m \in \mathbb{R}^5$, respectively, by means of the following torque distribution transformation:

$$\tau = \begin{bmatrix} J_b & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \tau_b \\ \tau_m \end{bmatrix} \tag{26}$$

where J_b depends on the vehicle's parameters. We computed the d'Alembert-Lagrange equation (10) with the spatial and recursive formulation of Featherstone (2010).

5.1 The structure of joint limits and reaching tasks

We considered two hierarchical tasks in this work. The primary task is dedicated to joint limits activation, and the secondary one consists of reaching a desired position and orientation with the robot's end-effector. It is important to point out that the joint limits avoidance task is naturally defined as a set of inequalities that must be satisfied along the motion. In the proposed scheme we handled inequalities by means of smooth transition functions to activate or inactivate the joint limits. In particular, the controller monitors the behavior of robot's joint profiles to detect the instant at which one or more joint coordinates reach either their lower or upper limit. When this happens, a joint limits task is activated at the primary hierarchical level. The task for the upper limit of the j-th joint coordinate is:

$$e_j = \tilde{q}_j - q_j \tag{27}$$

where $\tilde{q}_j = \bar{q}_j - \beta$ is simply the difference between the upper joint limit $\bar{q}_j$ and the activation buffer β. In this case, the second order task dynamics is:

$$\ddot{e}_j = J_{l_j} A^{-1} u_{q_{l_j}} + \mu_{l_j}$$

where $\mu_{l_j} = -J_{l_j} A^{-1} b$ with

$$J_{l_j} = [\alpha_0 \cdots \alpha_j \cdots \alpha_n] \in \mathbb{R}^{1 \times n}, \tag{28}$$

the element $\alpha_j = 1$ if $q_j \geq \tilde{q}_j$, i.e. the joint limit is an active constraint, otherwise $\alpha_j = 0$. The remaining row elements $\alpha_i = 0$ for $i = 1, \ldots, n$ and $i \neq j$. The control law for a single joint limit task j is computed as follows:

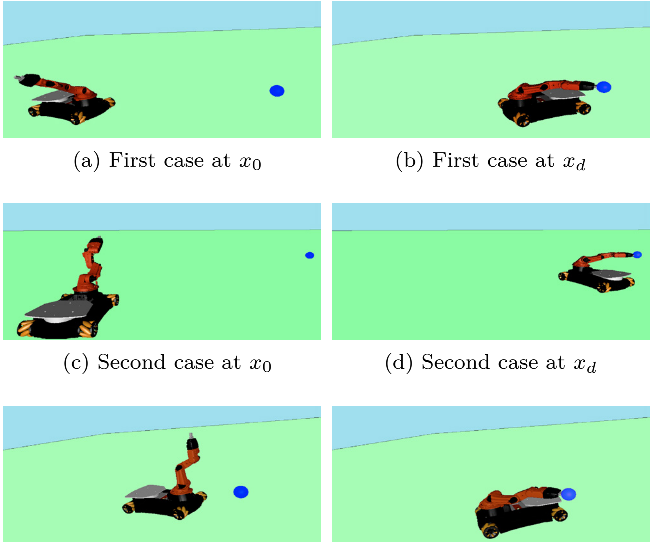
$$\begin{aligned} u_{q_{l_j}} &= \left(J_{l_j} A^{-1}\right)^{\#_A} \left(u_{l_j} - \mu_{l_j}\right) \\ u_{l_j} &= -k_{p_j} e_j - k_{d_j} \dot{q}_j \end{aligned} \tag{29}$$

where $u_{l_j} = -k_{p_j} e_j - k_{d_j} \dot{q}_j$ is the task space control input, k_{p_j} and k_{d_j} are positive constant gains.

The positioning task is defined as in (12):

$$e_r = \begin{bmatrix} e_p \\ e_Q \end{bmatrix} \tag{30}$$

where $e_p = x_p(q) - x_d \in \mathbb{R}^3$ is the error between the end-effector position and the desired one. We parametrized the end-effector's orientation with unit quaternions to avoid singularities. Thus, the task error with respect to the orientation of the robot's end-effector is:


(a) First case at x_0 (b) First case at x_d

(c) Second case at x_0 (d) Second case at x_d

(e) Third case at x_0 (f) Third case at x_d

Figure 1. **Robot initial and final postures for the three different cases.** The initial robot's end-effector is not the same while the desired position and orientation remain the same for the three cases

$$e_Q = Q_{d_1} Q_{[2:4]} + Q_1 Q_{d_{[2:4]}} - \widehat{Q_{d_{[2:4]}}} Q_{[2:4]} \tag{31}$$

where $Q_d = [Q_{d_1}\ Q_{d_2}\ Q_{d_3}\ Q_{d_4}]^T \in \mathbb{S}^3$ is the desired robot's end-effector orientation expressed in terms of a unit quaternion, $Q = [Q_1\ Q_2\ Q_3\ Q_4]^T \in \mathbb{S}^3$ is the current robot's end-effector orientation, and

$$\widehat{Q_{d_{[2:4]}}} = \begin{bmatrix} 0 & -Q_{d_4} & Q_{d_3} \\ Q_{d_4} & 0 & -Q_{d_2} \\ -Q_{d_3} & Q_{d_2} & 0 \end{bmatrix} \tag{32}$$

is the 3 by 3 skew-symmetric matrix operator.

The structure of the control law with joint limits and the positioning tasks is:

$$\begin{aligned} u_q &= u_{q_l} + u_{q_r} \\ u_{q_l} &= \left(J_l A^{-1}\right)^{\#_A} (u_l - \mu_l) \\ u_{q_r} &= \left(J_r A^{-1} N_l\right)^{\#_A} \left(u_r - \mu_r - J_r A^{-1} u_{q_l}\right) \end{aligned} \tag{33}$$

where $J_r = \dfrac{\partial e_r}{\partial q}$, $N_l = I_n - \left(J_l A^{-1}\right)^{\#_A} J_l A^{-1}$ is the null-space projector of the joint limits differential mapping $J_l A^{-1}$, u_l is the vector of control inputs for active joint limits (29), and $u_r = [u_x\ u_y\ u_z\ u_{Q_1}\ u_{Q_2}\ u_{Q_3}]^T$ is the vector of the predefined-time controllers (6) associated to robot's end-effector position and orientation coordinates.

Table 1. Initial and desired pose coordinates

	Case 1	Case 2	Case 3	Desired pose
p	0.1430	-1	1.2599	1
	-0.3490	-0.6249	1.5	2
	0.4465	0.5140	0.5805	0.3
Q	0.8001	0.6124	1	0.7071
	0.4619	-0.3536	0	0
	0.1913	0.3536	0	0.7071
	-0.3314	0.6124	0	0

5.2 Simulations for three different initial conditions

Three cases with different initial conditions were performed. The positioning task consists of reaching a desired position and orientation for the robot's end-effector as it is depicted in Figure 1. The blue sphere illustrates the desired position to be reached by the robot's end-effector. In Table 1 the desired position and orientation in terms of unit quaternions is specified. The desired values for the positioning tasks remain the same for three different initial conditions (i.e. the initial position and orientation of the robot's end-effector).

In Table 1, $p \in \mathbb{R}^3$ and $Q \in \mathbb{S}^3$ encode the position and orientation respectively. Position quantities are given in meters. The predefined-time convergence was set to 4 seconds, with $k_{fr} = 30$, $k_1 = 8$ and $k_2 = 6$. The time basis vector $g(t)$ with $r = 5$ was used to compute the TBG functions $h(t)$, $\dot{h}(t)$ and $\ddot{h}(t)$. The joint limits gains were set to $k_{p_j} = 50$ and $k_{d_j} = 40$, and the buffer β_i was 10% of the total j-th joint motion range. The parametric uncertainties to compute the robot's equations of motion were set to 0.8 times the real parameters of A. In addition, a time-varying perturbation is considered by means of a force profile acting at the robot's end-effector $\rho(t) = J_r^T f_\rho(t)$, where $f_\rho(t) = 2\sin(t)\,N$.

The positioning errors of all task coordinates converge at the predefined time despite different initial conditions, joint limit activation and disturbances as it is depicted in Figure 2. In Figure 2.(d) the behavior of a PD-like controller is illustrated for comparison purposes. As it is observed, the PD-like controller was unable to overcome disturbances.

The profiles of the auxiliary control inputs in u_r are shown in Figure 3. For the first and second cases, the control signals converge to zero at $4s$. However, for the third case in Figure 3.(c) the control signals must compensate the existence of bounded disturbances. The corresponding auxiliary control profiles for the PD-like controller are depicted in Figure 3.(d). In addition, the sliding surfaces are shown in Figure 4.

In Figure 5 we depicted the robot's torque profiles with the corresponding magnitudes. It can be seen that the torque profiles generated with the proposed controller did not surpass the capability of the real robot ($20Nm$). In contrast, the exponential convergence behavior of the PD-like controller created an important amount of torque magnitudes, surpassing the robot's capability.

The primary task was dedicated to joint limits avoidance in (33). Thus, in Figure 6 is shown the behavior of two active joint limits. The activation and deactivation are handled with smooth transition functions that led to smooth joint profiles. It is important to note that active joint limits did not affect the control performance.

6. CONCLUSIONS

In this paper, we have proposed a control scheme that achieves closed loop convergence in a preset time. It has been applied for a class of nonlinear second order systems. The control scheme guarantees the robust reference tracking of signals given by time base generators (TBGs), which determine the convergence time. We have taken advantage of the robustness of our controller to extend its application to solve problems in the hierarchical task-based inverse dynamics framework, where a second order system must be controlled for each task. Real applications of the hierarchical task-based control are usually affected by parametric uncertainties in the robot model and external disturbances, which makes necessary the use of robust controllers. This is tacked in this work by introducing a term given by a super-twisting control to ensure tracking of the TBGs. The proposed control scheme guarantees that each task converges to its desired value in the predefined time independently on the robot's initial state. If two or more tasks are in conflict, then the tasks with lower hierarchy tend to converge without altering the convergence of higher hierarchical tasks. The complete scheme has demonstrated a good performance in simulations using a kinematically redundant mobile-manipulator.

ACKNOWLEDGEMENTS

The third and fourth authors have been partially supported by CONACyT with grants 220796 and 285599, respectively.

REFERENCES

Becerra, H.M., Vázquez, C.R., Arechavaleta, G., and Delfin, J. (2017). Predefined-time convergence control for high-order integrator systems using time base generators. *IEEE Trans. on Control Systems Tech.*

Bhat, S.P. and Bernstein, D.S. (1998). Continuous finite-time stabilization of the translational and rotationnal double integrator. *IEEE Trans. on Automatic Control*, 43(5), 678–682.

Chang, P.H. and Jeong, J.W. (2012). Enhanced operational space formulation for multiple tasks by using time-delay estimation. *IEEE Trans. on Robotics*, 28(4), 773–786.

Featherstone, R. (2010). A beginner's guide to 6-d vectors (part 1). *IEEE Robotics & Automation Magazine*, 17(3), 83–94.

Jimenez-Rodriguez, E., Sanchez-Torres, J.D., Gomez-Gutierrez, D., and Loukianov, A.G. (2017). Predefined-time stabilization of high order systems. In *American Control Conference*, 5836–5841.

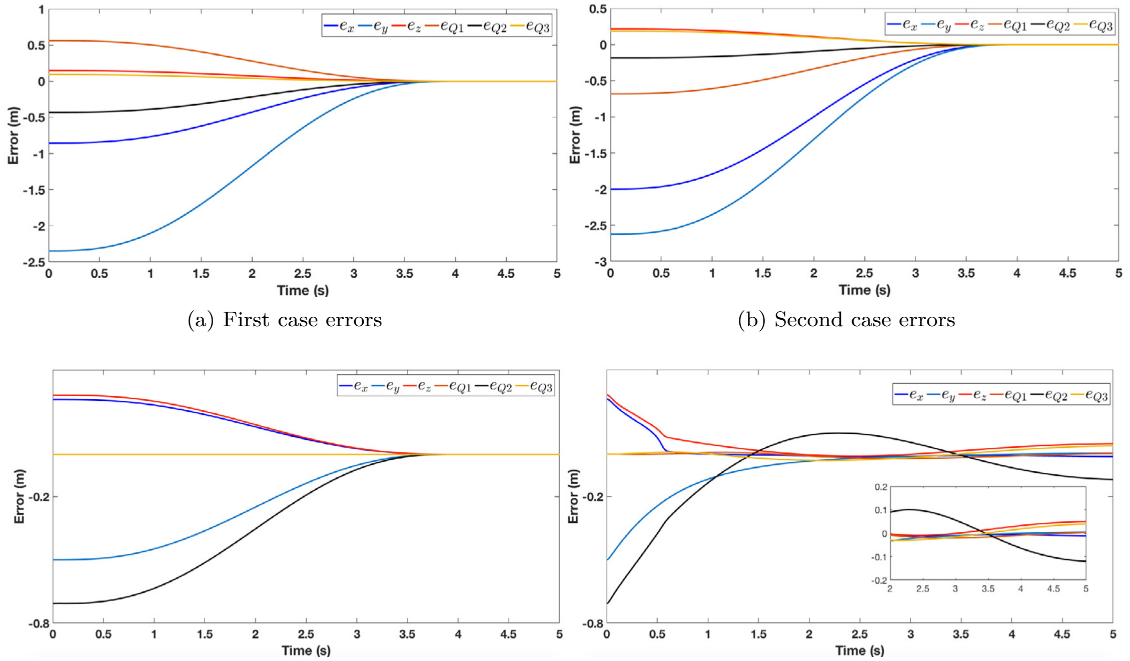
Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. of Robotics and Automation*, 3(1), 43–53.

Levant, A. (2013). On fixed and finite time stability in sliding mode control. In *IEEE Conf. on Decision and Control*, 4260–4265.

Moreno, J.A. and Osorio, M. (2012). Strict Lyapunov functions for the super-twisting algorithm. *IEEE Trans. on Automatic Control*, 57(4), 1035–1040.

Nakanishi, J., Cory, R., Mistry, M., Peters, J., and Schaal, S. (2008). Operational space control: A theoretical and empirical comparison. *The Int. J. of Robotics Research*, 27(6), 737–757.

Polyakov, A. (2012). Nonlinear feedback design for fixed-time stabilization of linear control systems. *IEEE Trans. on Automatic Control*, 57(8), 2106–2110.



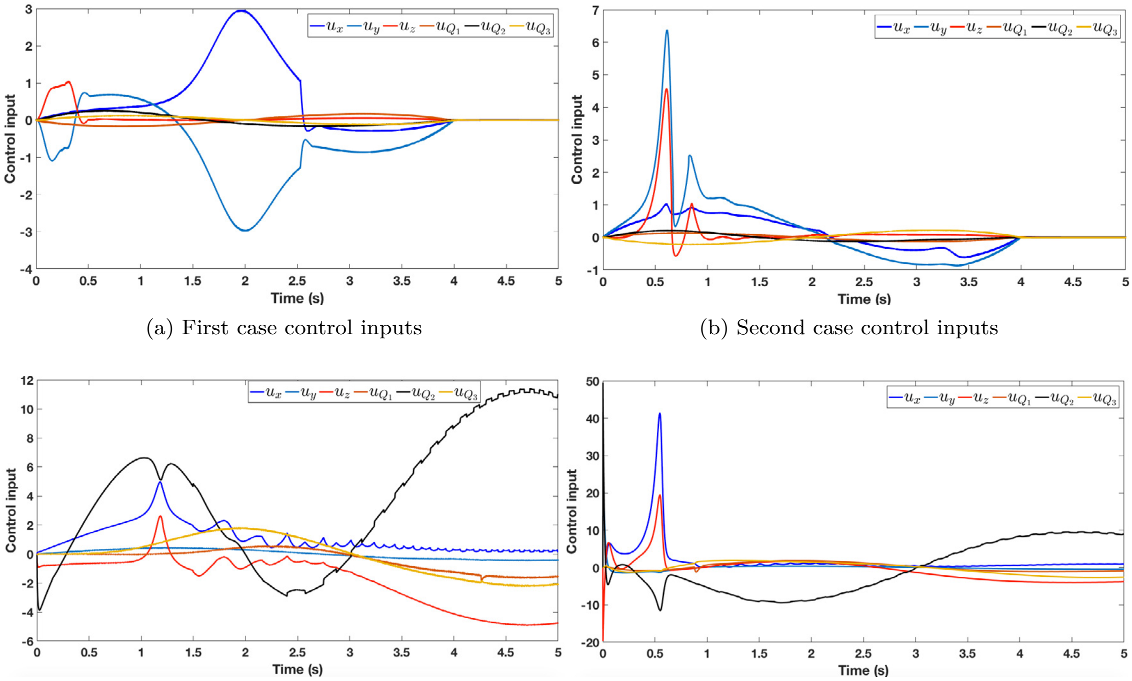
(a) First case errors

(b) Second case errors

(c) Third case errors

(d) Third case errors with PD controller

Figure 2. **Convergence of the task errors at the predefined time** $4s$. For the first and second cases (a), (b), respectively, it is assumed perfect knowledge of model dynamics. The third case in (c) considers model uncertainties and bounded matched perturbations. The same initial condition for the third case is used in (d), but with a PD-like controller.



(a) First case control inputs

(b) Second case control inputs

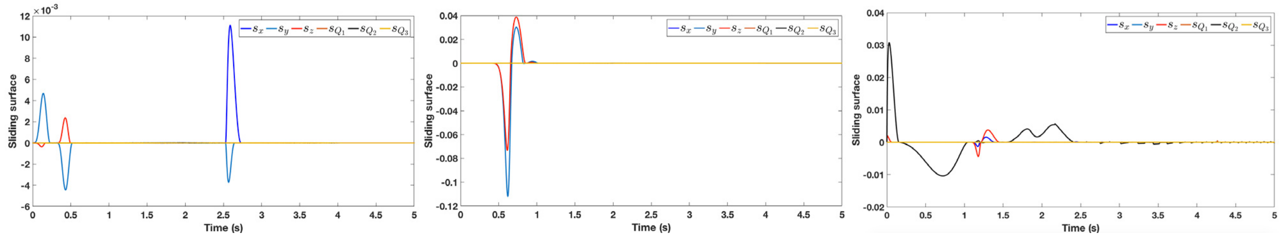
(c) Third case control inputs

(d) Third case control inputs with PD control

Figure 3. **The auxiliary control input profiles at predefined time** $4s$. (a) and (b) without additional disturbances (perfect knowledge of system dynamics). Proposed controller overcomes disturbances (c). PD-like control behavior in the presence of disturbances (d)

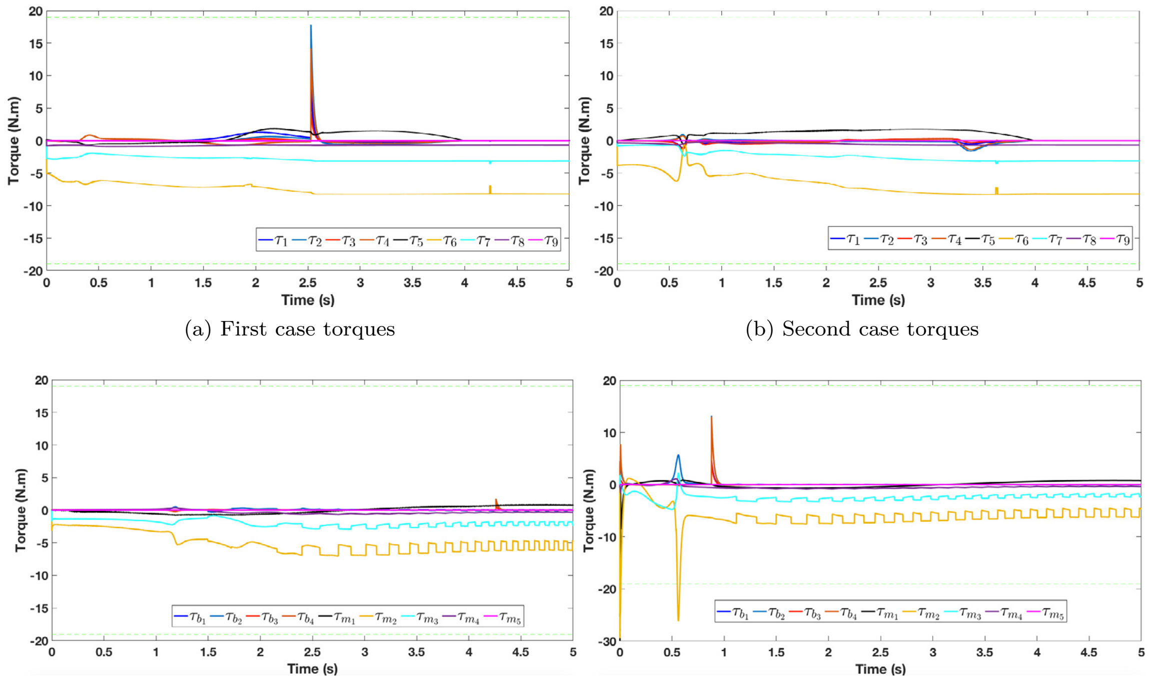
Polyakov, A., Efimov, D., and Perruquetti, W. (2015). Finite-time and fixed-time stabilization: Implicit lyapunov function approach. *Automatica*, 51(2015), 332–340.

Saab, L., Ramos, O.E., Keith, F., Mansard, N., Soueres, P., and Fourquet, J.Y. (2013). Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Trans. on Robotics*, 29(2), 346–362.



(a) Sliding surfaces for the first case (b) Sliding surfaces for the second case (c) Sliding surfaces for the third case

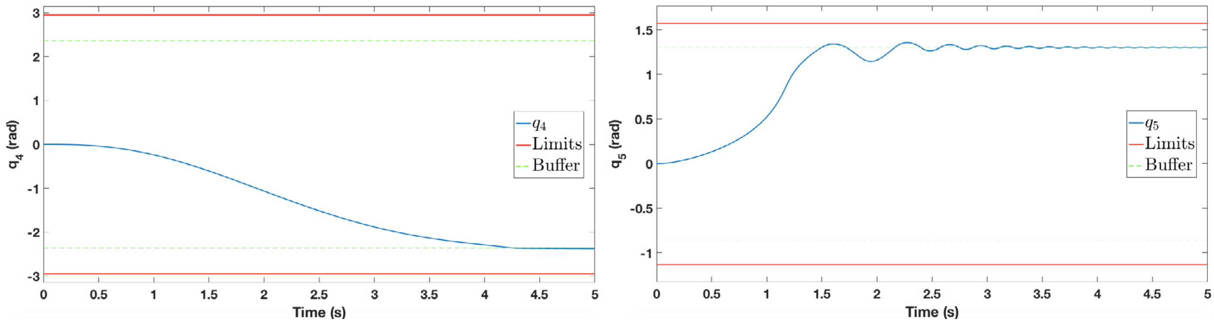
Figure 4. **Sliding surfaces for the positioning task** (a) and (b) without the presence of disturbances. In the presence of disturbances (c)



(a) First case torques (b) Second case torques

(c) Third case torques (d) Third case torques with PD control

Figure 5. **Joint space torque inputs.** (a) and (b) correspond to the first and second cases without disturbances. For comparison purposes the torque inputs for the third case in the presence of disturbances are depicted with the proposed (c) and PD-like (d) controllers, respectively.



(a) Third case with joint limit 4 (b) Third case with joint limit 5

Figure 6. **Activate joint limits for the third case.**

Samson, C., Borgne, M.L., and Espiau, B. (1991). *Robot Control: The Task Function Approach*, volume 22 of *Oxford Engineering Science Series*. Oxford University Press, New York, USA, first edition.

Siciliano, B. and Slotine, J.J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *IEEE International Conference on Advanced Robot*, 1211–1216. Pisa, Italy.

Torres, J.D.S., Sanchez, E.N., and Loukianov, A.G. (2015). Predefined-time stability of dynamical systems with sliding modes. In *American Control Conference*, 5842–5846.